



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/709,522

05/11/2004

Krishna Mohan ITIKARLAPALLI

ORCL-003

3521

51121

7590

05/28/2008

LAW FIRM OF NAREN THAPPETA  
158, PHASE ONE PALM MEADOWS, RAMAGUNDANAHALLI  
AIRPORT WHITEFIELD ROAD  
BANGALORE, 560066  
INDIA

EXAMINER

SANDERS, AARON J

ART UNIT

PAPER NUMBER

2168

MAIL DATE

DELIVERY MODE

05/28/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/709,522	<b>Applicant(s)</b> ITIKARLAPALLI ET AL.	
	<b>Examiner</b> AARON SANDERS	<b>Art Unit</b> 2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 12 February 2008.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-10, 13-21 and 25 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-10, 13-21 and 25 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)          | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

## **DETAILED ACTION**

### ***Response to Amendment***

Applicant's amendment filed 12 February 2008 has been entered. Claims 1-10, 13-21 and 25 are pending. Claims 1, 7, 10, 13-14 and 16 are currently amended. Claims 11-12 and 22-24 are cancelled. Claim 25 is new. This action is FINAL, as necessitated by amendment.

### ***Specification***

Applicant's amendment to the specification filed 12 February 2008 has been entered.

The abstract of the disclosure is objected to because it contains a paragraph number. The sheet presenting the abstract may not include other parts of the application or other material. See 37 C.F.R. 1.72.

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gostanian et al., U.S. 5,781,910 (Gostanian), in view of "XA Components," Oracle® JDBC Developer's Guide and Reference, 1999 (Oracle).

1. Gostanian teaches “A method of implementing atomic transactions in a system, said method comprising,” see Fig. 5 and col. 13, lines 41-47, “FIG. 5 is a highly schematized flow diagram of a 2PCC protocol 500.”

Gostanian teaches “requesting in a user program a transaction identifier for an atomic transaction,” see Figs. 3, 5, col. 9, lines 1-21, “Each application client 302-308 is essentially an application program that preferably resides on a client computer 220 (FIG. 2),” col. 9, lines 27-42, “The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308” and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction,” where the claimed “user program” is the referenced “application program.”

Gostanian teaches “generating said transaction identifier in a transaction manager in response to said requesting,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction.”

Gostanian teaches “executing a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “The worker process 522 of the cohort 514 then issues a set of update commands, as shown by block 532, to the corresponding database server (not shown), directing the database server to perform the operations of the requested transaction,” where the claimed “set of task procedures” is the referenced “set of update commands.”

Gostanian teaches “and executing said set of rollback procedures in a reverse order of said sequential order if said atomic transaction is to be aborted,” see Fig. 5 and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560.”

Gostanian teaches “wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program,” see col. 1, lines 40-62, “If a transaction is aborted, then any partial results (i.e., updates from those operations that were performed prior to the abort decision) must be undone. This process of returning the data items to their original values is also referred to as a roll back.”

Gostanian teaches “keeping track of a set of rollback procedures corresponding to said set of task procedures, each of said set of procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures,” see Fig. 5, col. 13, line 61 – col. 14, line 9, “The worker process 522 of the cohort 514 then issues a set of update commands, as shown by block 532, to the corresponding database server (not shown), directing the database server to perform the operations of the requested transaction” and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560,” where the claimed “corresponding” is that referenced “updates” and “roll back” are executed by the “worker.” Gostanian does not explicitly teach “said combination being contained in said plurality of combinations specified in said user program.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “The

Art Unit: 2168

transaction manager uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction,” where the claimed “plurality of combinations” are the referenced “XA resource instances.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

Gostanian does not explicitly teach “specifying in said user program a plurality of combinations, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “An XA resource instance is an instance of a class that implements the standard `javax.transaction.xa.XAResource` interface: `public interface XAResource { ... void rollback(Xid xid) throws XAException; void start(Xid xid, int flags) throws XAException; ... }`,” where the claimed “user program” is the referenced “instance,” the claimed “combinations” are included in the referenced “instance,” the claimed “transaction identifier” is the referenced “xid,” the claimed “task procedure” is the referenced “start” and the claimed “rollback procedure” is the referenced “rollback.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between

the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

2. Gostanian teaches “The method of claim 1, wherein said transaction identifier is unique to each of the atomic transactions,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction.”

5. Gostanian teaches “The method of claim 1, further comprising examining a status returned by execution of one of said task procedures and performing said aborting if said status indicates an error,” see Fig. 5, col. 14, lines 18-42, “The status request message 542 inquires whether each cohort 514 is ready and able to commit the transaction” and col. 14, line 43-63, “The coordinator 512 then transmits a final decision message 552 to the cohorts 514 instructing them to either commit or abort the transaction.”

6. Gostanian teaches “The method of claim 1, wherein said aborting is performed asynchronously,” see Fig. 5 and col. 14, line 64 - col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560.”

7. Gostanian teaches “A computer readable medium carrying one or more sequences of instructions representing a user program for execution on a system, said user program implementing an atomic transaction, wherein execution of said one or more sequences of instructions by one or more processors contained in said system causes said one or more processors to perform the actions of,” see Fig. 5 and col. 13, lines 41-47, “FIG. 5 is a highly schematized flow diagram of a 2PCC protocol 500.”

Gostanian teaches “requesting an identifier in said user program from a transaction manager for said atomic transaction, wherein said transaction manager generates a unique value as said identifier,” see Figs. 3, 5, col. 9, lines 1-21, “Each application client 302-308 is essentially an application program that preferably resides on a client computer 220 (FIG. 2),” col. 9, lines 27-42, “The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308” and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction,” where the claimed “user program” is the referenced “application program.”

Gostanian teaches “and aborting said atomic transaction by specifying said identifier associated with an abort procedure to cause said rollback procedures to be executed,” see Fig. 5 and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560.”

Gostanian does not teach “setting a variable to equal said identifier.” Oracle does, however, see “XA ID Interface and Oracle Implementation,” “An XA transaction ID instance is an instance of a class that implements the standard javax.transaction.xa.Xid interface, which is a Java mapping of the X/Open transaction identifier XID structure.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”



Gostanian does not explicitly teach “specifying a plurality of combinations in said user program for execution in said system, wherein each of said plurality of combinations contains said variable, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said variable in each of said plurality of combinations specifies said identifier generated by said transaction manager.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “An XA resource instance is an instance of a class that implements the standard `javax.transaction.xa.XAResource` interface: public interface `XAResource` { ... void rollback(Xid xid) throws XAException; void start(Xid xid, int flags) throws XAException; ... },” where the claimed “user program” is the referenced “instance,” the claimed “combinations” are included in the referenced “instance,” the claimed “transaction identifier” is the referenced “xid,” the claimed “task procedure” is the referenced “start” and the claimed “rollback procedure” is the referenced “rollback.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

Gostanian does not teach “wherein said plurality of combinations and said abort procedure are contained in a said user program.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “The transaction manager uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction,” where the claimed “plurality of combinations” are the referenced “XA resource instances.” Thus, it would

Art Unit: 2168

have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle's teachings would have allowed Gostanian's method to coordinate distributed transactions between the transaction manager and all the transaction branches, see "XA Resource Interface and Oracle Implementation."

8. Gostanian does not teach "The computer readable medium of claim 7, wherein said specifying comprises including each of said plurality of combinations in a single procedure call." Oracle does, however, see "XA Resource Interface and Oracle Implementation," "The transaction manager uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction," where the claimed "plurality of combinations" are the referenced "XA resource instances." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle's teachings would have allowed Gostanian's method to coordinate distributed transactions between the transaction manager and all the transaction branches, see "XA Resource Interface and Oracle Implementation."

9. Gostanian teaches "The computer readable medium of claim 7, further comprising examining a status returned by execution of one of said task procedures and performing said aborting if said status indicates an error," see Fig. 5, col. 14, lines 18-42, "The status request message 542 inquires whether each cohort 514 is ready and able to commit the transaction" and col. 14, line 43-63, "The coordinator 512 then transmits a final decision message 552 to the cohorts 514 instructing them to either commit or abort the transaction."

10. Gostanian teaches "A computer readable medium carrying one or more sequences of instructions for supporting implementation of an atomic transaction in a system, wherein

Art Unit: 2168

execution of said one or more sequences of instructions by one or more processors contained in said system causes said one or more processors to perform the actions of,” see Fig. 5 and col. 13, lines 41-47, “FIG. 5 is a highly schematized flow diagram of a 2PCC protocol 500.”

Gostanian teaches “generating an identifier for said atomic transaction for a user program,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction.”

Gostanian teaches “executing a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said plurality of combinations,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “The worker process 522 of the cohort 514 then issues a set of update commands, as shown by block 532, to the corresponding database server (not shown), directing the database server to perform the operations of the requested transaction,” where the claimed “set of task procedures” is the referenced “set of update commands.”

Gostanian teaches “and executing said set of rollback procedures in a reverse order of said sequential order in response to receiving an abort request,” see Fig. 5 and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560.”

Gostanian teaches “wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program,” see col. 1, lines 40-62, “If a transaction is aborted, then any partial results (i.e., updates from those operations that were performed prior to

the abort decision) must be undone. This process of returning the data items to their original values is also referred to as a roll back.”

Gostanian teaches “keeping track of a set of rollback procedures corresponding to said set of task procedures, each of said set of procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures,” see Fig. 5, col. 13, line 61 – col. 14, line 9, “The worker process 522 of the cohort 514 then issues a set of update commands, as shown by block 532, to the corresponding database server (not shown), directing the database server to perform the operations of the requested transaction” and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560,” where the claimed “corresponding” is that referenced “updates” and “roll back” are executed by the “worker.” Gostanian does not explicitly teach “said combination being contained in said plurality of combinations specified in said user program.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “The transaction manager uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction,” where the claimed “plurality of combinations” are the referenced “XA resource instances.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

Gostanian does not explicitly teach “receiving a plurality of combinations for execution from said user program, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “An XA resource instance is an instance of a class that implements the standard javax.transaction.xa.XAResource interface: public interface XAResource { ... void rollback(Xid xid) throws XAException; void start(Xid xid, int flags) throws XAException; ... },” where the claimed “user program” is the referenced “instance,” the claimed “combinations” are included in the referenced “instance,” the claimed “transaction identifier” is the referenced “xid,” the claimed “task procedure” is the referenced “start” and the claimed “rollback procedure” is the referenced “rollback.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

13. Gostanian teaches “The computer readable medium of claim 10, wherein said transaction identifier is generated to be unique for each atomic transaction,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction.”

16. Gostanian teaches “A computer system comprising,” see Fig. 2 and col. 7, lines 46-62, “FIG. 2 is a block diagram of a database system 200.”

Gostanian teaches “a memory storing a plurality of instructions,” see Fig. 2 and col. 8, lines 14-24, “As shown in FIG. 2, a typical hardware configuration of a client 220 includes a central processing unit (CPU) 222 coupled between a memory 224.”

Gostanian teaches “and a processing unit coupled to said memory and executing said plurality of instructions to support implementation of atomic transactions in a programming environment, said processing unit being operable to,” see Fig. 2 and col. 8, lines 14-24, “As shown in FIG. 2, a typical hardware configuration of a client 220 includes a central processing unit (CPU) 222 coupled between a memory 224.”

Gostanian teaches “request in a user program a transaction identifier for an atomic transaction,” see Figs. 3, 5, col. 9, lines 1-21, “Each application client 302-308 is essentially an application program that preferably resides on a client computer 220 (FIG. 2),” col. 9, lines 27-42, “The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308” and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction,” where the claimed “user program” is the referenced “application program.”

Gostanian teaches “generate said transaction identifier in a transaction manager in response to said requesting, wherein said transaction manager is provided external to said user program,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG.

4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction.”

Gostanian teaches “execute a set of task procedures in a sequential order according to said user program,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “The worker process 522 of the cohort 514 then issues a set of update commands, as shown by block 532, to the corresponding database server (not shown), directing the database server to perform the operations of the requested transaction,” where the claimed “set of task procedures” is the referenced “set of update commands.”

Gostanian teaches “and execute said set of rollback procedures in a reverse order of said sequential order if said atomic transaction is to be aborted, wherein said rollback procedures are identified according to said keeping,” see Fig. 5 and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560.”

Gostanian teaches “keep track of a set of rollback procedures corresponding to said set of task procedures, each of said set of procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures,” see Fig. 5, col. 13, line 61 – col. 14, line 9, “The worker process 522 of the cohort 514 then issues a set of update commands, as shown by block 532, to the corresponding database server (not shown), directing the database server to perform the operations of the requested transaction” and col. 14, line 64 – col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560,” where the claimed “corresponding” is that referenced “updates”

and “roll back” are executed by the “worker.” Gostanian does not explicitly teach “said combination being contained in said plurality of combinations specified in said user program.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “The transaction manager uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction,” where the claimed “plurality of combinations” are the referenced “XA resource instances.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

Gostanian does not explicitly teach “specify in said user program a plurality of combinations for execution in a sequential order, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said rollback procedure is specified as a separate procedure from said task procedure.” Oracle does, however, see “XA Resource Interface and Oracle Implementation,” “An XA resource instance is an instance of a class that implements the standard javax.transaction.xa.XAResource interface: public interface XAResource { ... void rollback(Xid xid) throws XAException; void start(Xid xid, int flags) throws XAException; ... },” where the claimed “user program” is the referenced “instance,” the claimed “combinations” are included in the referenced “instance,” the claimed “transaction identifier” is the referenced “xid,” the claimed “task procedure” is the referenced “start” and the claimed “rollback



procedure” is the referenced “rollback.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Oracle’s teachings would have allowed Gostanian’s method to coordinate distributed transactions between the transaction manager and all the transaction branches, see “XA Resource Interface and Oracle Implementation.”

17. Gostanian teaches “The computer system of claim 16, wherein said transaction identifier is unique to each of the atomic transactions,” see Fig. 5 and col. 13, line 61 – col. 14, line 9, “As with the 1PPC protocol 400 (FIG. 4), a manager process 516 of the coordinator 512 first assigns a unique transaction identification code 524 to the particular transaction.”

20. Gostanian teaches “The computer system of claim 16, wherein said processing unit is further operable to examine a status returned by execution of one of said task procedures and to perform said aborting if said status indicates an error,” see Fig. 5, col. 14, lines 18-42, “The status request message 542 inquires whether each cohort 514 is ready and able to commit the transaction” and col. 14, line 43-63, “The coordinator 512 then transmits a final decision message 552 to the cohorts 514 instructing them to either commit or abort the transaction.”

21. Gostanian teaches “The computer system of claim 16, wherein said processing unit is operable to execute said rollback procedures asynchronously,” see Fig. 5 and col. 14, line 64 - col. 15, line 10, “If the decision message 552 is an abort, the cohort 514 aborts the transaction and the worker process 522 directs the corresponding database server to roll back the results as shown by block 560.”

25. Gostanian teaches “The computer readable medium of claim 7, wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program,”

see col. 1, lines 40-62, “If a transaction is aborted, then any partial results (i.e., updates from those operations that were performed prior to the abort decision) must be undone. This process of returning the data items to their original values is also referred to as a roll back.”

Claims 3-4, 14-15 and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gostanian et al., U.S. 5,781,910 (Gostanian), in view of “XA Components,” Oracle8i JDBC Developer’s Guide and Reference, 1999 (Oracle), and further in view of Raz, U.S. 5,701,480 (Raz).

3. Gostanian and Oracle do not teach “The method of claim 1, wherein said keeping comprises storing data representing said rollback procedures in a stack.” Raz does, however, see col. 19, lines 51-59, “the transaction scheduler responds to an interrupt by removing the context of the interrupted transaction from the processor stack of the digital computer... The context includes the value of the program counter which points to the interrupted memory location in the transaction program.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz’s teachings would have allowed Gostanian and Oracle’s method to gain way to keep track of transactions, see col. 19, lines 51-59.

4. Gostanian and Oracle do not teach “The method of claim 3, wherein said stack is stored in a memory.” Raz does, however, see col. 2, lines 7-24, “the operating system typically provides an established set of memory management procedures that can be invoked or called from an application program to define a ‘recovery unit’”, where the “stack” in the reference is part of the “recovery unit.” Thus, it would have been obvious to one of ordinary skill in the

Art Unit: 2168

database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed Gostanian and Oracle's method to gain way to keep track of transactions, see col. 19, lines 51-59.

14. Gostanian and Oracle do not teach "The computer readable medium of claim 10, wherein said set of rollback procedures are represented in the form of a stack." Raz does, however, see col. 19, lines 51-59, "the transaction scheduler responds to an interrupt by removing the context of the interrupted transaction from the processor stack of the digital computer... The context includes the value of the program counter which points to the interrupted memory location in the transaction program." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed Gostanian and Oracle's method to gain way to keep track of transactions, see col. 19, lines 51-59.

15. Gostanian and Oracle do not teach "The computer readable medium of claim 14, wherein said stack is stored in a memory." Raz does, however, see col. 2, lines 7-24, "the operating system typically provides an established set of memory management procedures that can be invoked or called from an application program to define a 'recovery unit'", where the "stack" in the reference is part of the "recovery unit." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed Gostanian and Oracle's method to gain way to keep track of transactions, see col. 19, lines 51-59.

18. Gostanian and Oracle do not teach "The computer system of claim 16, wherein said processing unit is operable to store data representing said rollback procedures in a stack to

Art Unit: 2168

perform said keep.” Raz does, however, see col. 19, lines 51-59, “the transaction scheduler responds to an interrupt by removing the context of the interrupted transaction from the processor stack of the digital computer... The context includes the value of the program counter which points to the interrupted memory location in the transaction program.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz’s teachings would have allowed Gostanian and Oracle’s method to gain way to keep track of transactions, see col. 19, lines 51-59.

19. Gostanian and Oracle do not teach “The computer system of claim 18, wherein said stack is stored in a memory.” Raz does, however, see col. 2, lines 7-24, “the operating system typically provides an established set of memory management procedures that can be invoked or called from an application program to define a ‘recovery unit’”, where the “stack” in the reference is part of the “recovery unit.” Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz’s teachings would have allowed Gostanian and Oracle’s method to gain way to keep track of transactions, see col. 19, lines 51-59.

### ***Response to Arguments***

Applicant's arguments with respect to the 35 U.S.C. 102 rejection of claims 1-24 have been considered but are moot in view of the new ground(s) of rejection.

***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aaron J. Sanders whose telephone number is 571-270-1016. The examiner can normally be reached on M-Th 8:00a-5:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vo Tim can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2168

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tim T. Vo/  
Supervisory Patent Examiner, Art Unit  
2168

/Aaron Sanders/  
Examiner, Art Unit 2168  
14 May 2008

/S. P./  
Primary Examiner